REMARKS/ARGUMENTS

These remarks are made in response to the Office Action of June 1, 2005 (Office Action). This response is filed after the 3-month shortened statutory period, and as such, a retroactive extension of time is hereby requested. The Examiner is authorized to charge the appropriate extension fee to Deposit Account 50-0951.

Claims 1-5, 6, and 9-18 were rejected at page 2 of the Office Action under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,088,671 to Gould, *et al.* (hereinafter Gould). At page 5 of the Office Action, Claims 8 and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Gould in view of U.S. Patent No. 6,539,080 to Bruce, *et al.* (hereinafter Bruce). Claims 7 and 19 were also rejected at page 5 under 35 U.S.C. § 103(a) as being unpatentable over Gould.

I. Applicants' Invention

It may be useful to reiterate certain aspects of Applicants' invention prior to addressing the cited references. One aspect of the invention is the auditory presentation of database query results through an audio user interface (AUI). More particularly, the invention allows each choice extracted from a database query to be audibly presented immediately upon its extraction rather than according to conventional processes that extract all matches and, only then, presents them in batch. With Applicants' invention, a user can respond to each choice when that choice is presented, thereby interrupting the database query operation at any point so as to preclude further subsequent presentations of additional choices through the audio user interface (AUI).

One embodiment of the invention, typified by independent Claim 1, is a method for presenting database query results through an AUI. The method includes initiating a database query operation, which retrieves a database query result item from at least one database. The method further includes presenting each query result item through the AUI

BEST AVAILABLE COPY

as each query result item is found in the at least one database, the presenting step occurring concurrently with the database query operation.

II. <u>The Claims Define Over The Prior Art</u>

As already noted, Claims 1-5, 6, and 9-18 were rejected as being anticipated by Gould. Gould is directed to a system and method for recognizing speech and distinguishing between dictation and commands. (Col. 1, lines 35-36; Abstract.) With Gould, speech recognition is based on signals representing speech elements that include elements "corresponding to text to be recognized and command elements to be executed." (Col. 1, lines 36-39; Abstract.) Once every one of the speech elements are recognized in Gould, "the recognized elements are acted on in a manner which depends on whether [the elements] are text or commands." (Col. 1, lines 39-41; Abstract.)

Applicants respectfully maintain that Gould fails to expressly or inherently teach each aspect of Applicants' invention. Gould does not, for example, teach initiating a database query operation that retrieves a plurality of database query results from one or more databases, as expressly recited in independent Claims 1, 9, and 13. Rather Gould describes a dictation system the determines whether a user-supplied utterance should be interpreted as dictated text or as a speech command (i.e., not text to be dictated).

The distinction is seen in a portion of Gould referenced at page 7 of the Office Action. In this portion, Gould, in referring to Figure 4, states that:

> "once the vocabularies are stored in local memory an application calls the recognition software, [and] the CPU compares speech frames representing the user's speech to speech models in the vocabularies to recognize (step 60) the user's speech. The CPU then determines (steps 62 and 64) whether the results represent a command or text. Commands include single words and phrases and sentences that are defined by templates (i.e., restriction

rules). The templates define the words that may be said within command sentences and the order in which the words are spoken. The CPU compares (step 62) the recognition results to the possible command words and phrases and to command templates, and if the results match a command word or phrase or a command template (step 64), then the CPU sends (step 65a) the application that called the speech recognition software keystrokes or scripting language that cause the application to execute the command, and if the results do not match a command word or phrase or a command template, the CPU sends (step 65b) the application keystrokes or scripting language that cause the application to type the results as text." (Col. 4, lines 49-67.)

As the quoted portion demonstrates, Gould describes a dictation system in which a system determines whether a user's utterance should be treated as dictated text or as a speech command that is not to be dictated. If, for example, a user were to say "four score and seven years ago," then assuming perfect speech recognition, Gould's system would present the typed text FOUR SCORE AND SEVEN YEARS AGO because the text does not match a predefined, active command. Conversely, if the user were to say "Bold the previous three words," then assuming this matched an active command in a command list or grammar, the system would not type the words, but instead would bold SEVEN YEARS AGO.

Fundamentally, these activities described by Gould are peculiar to speech dictation systems; they are not database queries. In Gould, as in similar such systems, a user speaks a phrase, a speech recognizer continues accepting acoustic input until an end-of-speech detector determines that the user has stopped speaking, and then the speech recognizer determines the most-likely words based on the user utterance. The converting

the utterance to recognized words is based on (1) the recognizer's acoustic model and (2) the recognizer's language model.

The output of the recognizer is a text string, but this is typical of such dictation systems. Gould provides an extra step by delaying output of the string in the form of dictated text until after the system has determined whether the string matches an active command, perhaps one in a simple list of commands or a finite-state grammar that defines permissible commands. If the speech is recognized to be a command, then the command is executed. If not, the recognized words are produced as dictated text.

None of the described activities, however, are comparable to a database query, which by definition is able to return multiple matches. Gould's actions are carried out against text that is produced by a recognizer, not matches identified during a database query and returned from the database.

In a database query, a search string is submitted to a database, which has a wholly different structure than an acoustic model, a language model, and a finite-state grammar. A database query based on the search string, moreover, can produce no matches, a single match, or multiple matches. In the case of multiple matches, a user indicates which match is the desired one. Certainly, therefore, Applicants' invention does not involve nor require a speech recognition component that takes audio input, detects the end of the speech, and, based on an acoustic model and either a language model or a finite-state grammar, interprets the audio input so as to output a text string. In Gould, a text string is determined to be either dictated text or a command. By contrast, Applicants' invention utilizes a text string to perform a database query.

Another critical distinction is that, contrary to the results produced by a conventional database query in a speech application context, Applicants' invention precludes having to wait until all hits are returned from the database and then having to form a disambiguation prompt (e.g., "Was that Sam Hill in Raleigh or Sam Hill in Austin?"). Applicants' invention audibly presents database matches as they are returned,

the presenting occurring concurrently with the database query operation as recited in independent Claims 1, 9, and 13. Applicants' invention provides a mechanism whereby a spoken indication from a user indicates that a just-presented match is or is not the desired match.

As already noted, the actions described in Gould are those associated with the use of acoustic models, language models, and finite-state grammars. Indeed, Gould describes a comparison of the output of two recognizers: one for dictated text and one for predefined commands. By contrast, Applicants' invention can operate based on standard speech recognition elements, but unlike Gould or conventional systems, Applicants' invention presents matches from a database query that has nothing to do with the speech recognition. Applicants' invention, moreover, enables a user to confirm or disconfirm that a match as been identified through the query operation as soon as the matches are made. This is not expressly or inherently taught by Gould.

The speech processing in Gould is essentially an automatic determination of whether a text string produced by the recognizer should be interpreted as dictated text or a command. Gould presents a user no alternatives during a database query, only a determination as to whether a user input is dictated text or predefined command. Gould provides no mechanism by which a user is able to confirm or disconfirm matches concurrently with the operation of a database query.

The relevant literature supports the contention that the method in Gould is not a database lookup, but rather, speech processing according to operations divorced from the searching of databases. Two highly-regarded books concerning speech recognition/natural language processing are Frederick Jelinek, STATISTICAL METHODS FOR SPEECH RECOGNITION (MIT Press, 1997) and Christopher Manning and Hinrich Schutze, FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING (MIT Press, 1999). The index of the former discloses no entries for "database". The index of the later provides one entry for "database," and one for "database merging" (at p. 530). In the

Manning and Schutze text, the topic of database merging is included in the chapter on information retrieval, not in the chapters that describe the mechanics of speech recognition. By contrast, examination of the index of a standard text on VoiceXML programming, C. Sharma and J. Kunins, VoiceXML (Wiley Computer Publishing, 2002), reveals four independent entries for the term "database," including a lengthy description of how to work with databases in a case study of building an application.

The standard literature, accordingly, confirms that the usual programmer's conception of a database is that it is data structured into fields. A telephone directory is a database, typically including fields for a person's name and telephone number. Applications work with databases by adding new data, deleting old data, and looking up data. To look up data, one must submit a query which contains information about part of the record (name) in an attempt to retrieve some other part of the record (telephone number).

Speech recognition as in Gould operates in a manner wholly distinct from database query operations, as performed according to Applicants' invention. For example, database lookups are completely deterministic, whereas speech recognition processes as in Gould are inevitably probabilistic and usually based on hidden Markov models and/or other probabilistic models. In the Jelinek text referenced above, for example, the preface states that "[t]he text concentrates on those basic statistical ideas that have proven so fruitful in speech recognition: hidden Markov models, data clustering, smoothing of probability distributions, the decision tree method of equivalence classification, the use of information measures as goodness criteria, and maximum entropy probability estimation." The relevant literature and common usage thus confirm that Gould does not expressly or inherently teach a database query operation as recited in independent Claims 1, 9, and 13.

Yet another aspect of Applicants' invention is the presentation of "each query result item through the AUI" as explicitly recited in independent Claims 1, 9, and 13.

Applicants respectfully maintain that Gould also does not expressly or inherently teach this aspect of Applicants' invention. In the portion cited at page 7 of the Office Action, Gould states

> "The command browser displays possible commands for the application being executed. For example, a word processing application includes single command words, e.g., [Bold] 70 and [Center] 72, command phrases, e.g., [Close Document] 74 and [Cut This Paragraph] 76, and flexible sentence commands, e.g., [<Action><2 to 20><Text Objects>] 78 and [Move <Direction><2 to 20><Text Objects>] 80. Referring also to FIG. 6, the user may select a command shown in the command browser to display examples 82 of the selected command 80." (Col. 5, lines 6-16.)

Gould here describes the system's What-Can-I-Say user interface, which presents a partial list of the commands that the user can say at particular point in time. This, however, is not a list of commands that is created as a result of a user-initiated database query. It is a simple, predetermined list of commands that are valid for the current application. Accordingly, it is not a presentation of query result items through an AUI, as recited in independent Claims 1, 9, and 13.

Applicants' invention allows for the termination of a database query operation in response to a speech response. Applicants respectfully maintain that this feature is not expressly or inherently taught by Gould. In a portion cited at page 7 of the Office Action, Gould states that

> "[w]hile a user's speech is being recognized, the CPU sends keystrokes or scripting language to the application to cause the application to display partial results (i.e., recognized words within an utterance before the entire

> utterance has been considered) within the document being displayed on the display screen (or in a status window on the display screen). If the CPU determines that the user's speech is text and the partial results match the final results, then the CPU is finished. However, if the CPU determines that the user's speech was a command, then the CPU sends keystrokes or scripting language to the application to cause the application to delete the partial results from the screen and execute the command." (Col. 6, lines 19-34.)

Gould here describes a system in which, as the user speaks, the dictation application shows the recognizer's current best guess at the text that will ultimately be produced after end-of-speech detection. Based on the description, what Gould provides is thus an end-of-speech detection, not the termination of a database query that Applicants' invention provides. Applicants' invention provides that, using a speech command uttered during the presentment of database matches, a user can actively select one of the items returned from the database, as explicitly recited in Claims 2 and 14. Gould does not expressly or inherently teach such a feature.

With respect to Claim 9, specifically, Gould does not expressly or inherently teach a dialog manager that manages an audible presentation of database query results concurrently with database operations. In another portion noted at page 7 of the Office Action, Gould states

> "[i]nterrupt signal 26 also causes the operating system software to call monitor software 32. Monitor software 32 keeps a count 34 of the number of speech packets stored but not yet processed. An application 36, for example, a word processor, being executed by the CPU periodically checks for user input by examining the monitor software's count. If the count is

> zero, then there is no user input. If the count is not zero, then the application calls speech recognizer software 38 and passes a pointer 37 to the address location of the speech packet in buffer 30. The speech recognizer may be called directly by the application or may be called on behalf of the application by a separate program, such as DragonDictate.TM. from Dragon Systems.TM. of West Newton, Mass., in response to the application's request for input from the mouse or keyboard." (Col. 3, lines 36-49.)

Gould does not here teach, expressly or inherently, a dialog manager that manages the audible presentation of database query results concurrently with the database operation. Rather Gould is describing the monitoring of speech packets (speech input) that are monitored to determine whether an audio input is to be processed for the purpose of speech recognition. Gould, however, teaches nothing about the audible presentation of database query results.

Applicants respectfully maintain that Gould further fails to expressly or inherently teach using a text-to-speech processor as recited in dependent Claim 10. In another portion cited at page 7 of the Office Action, Gould states

> "As an alternative to dictating directly to an application, the user dictates text to a speech recognizer window, and after dictating a document, the user transfers the document (manually or automatically) to the application." (Col. 4, lines 12-15.)

This portion of Gould does not explicitly or inherently describe a text-to-speech processor. Instead, it describes the use of a special interface for holding the results of speech recognition (speech-to-text processing) before committing the recognized speech

to an application via manual or automatic transfer. Gould nowhere describes the use of text-to-speech (TTS) processing.

Applicants respectfully maintain that Gould does not expressly or inherently teach a "barge-in facility" as explicitly recited in dependent Claim 11. In the portion of Gould cited at page 7 of the Office Action, Gould states

> "[i]f the CPU determines that the user's speech is text and the partial results match the final results, then the CPU is finished. However, if the CPU determines that the user's speech is text but that the partial results do not match the final results, then the CPU sends keystrokes or scripting language to the application to correct the displayed text. Similarly, if the CPU determines that the user's speech was a command, then the CPU sends keystrokes or scripting language to the application to cause the application to delete the partial results from the screen and execute the command." (Col. 6, lines 24-34.)

Applicants respectfully maintain that Gould is only describing a dictation system that can distinguish between speech input intended to be produced as dictated text and speech input intended for interpretation as a command. It would not be useful to introduce or incorporate a "barge-in facility" into a dictation system. A dictation system must continuously listen for speech input. In Gould, the system is intended to produce some result for any and all speech input: either dictated text or a command. Because the system of Gould does not produce any speech output, there can be nothing into which a user would barge using a barge-in facility. Barge-in only makes sense in conversational systems – that is, systems that employ speech input and speech output – where the barge-in allows users to interrupt speech output. It follows that Gould can not be read as teaching a barge-in facility.

It is further asserted at page 8 of the Office Action that Bruce teaches that database query results can be presented through an AUI as the results are determined concurrently with the execution of a database operation. In the specific portion quoted at page 8 of the Office Action, Bruce states that

> "In a particular embodiment, the route to the destination location can be mapped taking into account the route traffic, travel-times, road conditions, and route weather conditions. The caller may receive the driving or route instructions in a variety of different ways. The route instructions can be communicated directly over the telephone from an interactive voice response system, a live operator, a synthesized voice, a voice mail message, and Internet electronic mail, an alpha/numeric pager or telephone or a Personal Digital Assistant ('PDA')." (Col. 2, lines 54-63.)

Although Bruce utilizes an AUI, there is not the slightest suggestion in Bruce of presenting the results of the database query as those results are obtained concurrently with the execution of the database operation. Applicants respectfully submit that Bruce is not only silent about the presentation of database matches as they are found, but, in fact, it would be a design mistake. It would be counterproductive to apply such a feature to Bruce because in a system presenting navigation instructions, a user would want instructions to either be available all at once so the user could write them down or, alternatively, to be presented on an as-needed basis driven by a user's requesting the next instruction or by having the application be aware of and responsive to the user's location. It logically follows that Bruce can not be read as inherently teaching the presentment of database query results through an AUI as the results are determined concurrently with the execution of a database operation, as recited in independent Claims 1, 9, and 13.

Applicants respectfully assert that Gould does not expressly or inherently teach every feature of independent Claims 1, 9, or 13, and the claims thus define over the prior art. Applicants further respectfully assert that Gould in combination with Bruce similarly fails to teach or suggest every feature recited in the claims. Applicants respectfully assert also that for the reasons stated herein, the additional features recited in Dependent Claims 2, 10, 11, 12, and 14 are likewise not expressly nor inherently taught by Gould, and thus these claims define over the prior art apart from the independent claims from which they depend. Applicants respectfully assert, moreover, that whereas Dependent Claims 3-8 and 15-20 each depend from one of the independent claims while reciting additional features, these claims likewise define over the prior art.

CONCLUSION

The Applicants believe that this application is now in full condition for allowance, which action is respectfully requested. The Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the application to completion.

Respectfully submitted,

Date: October 3, 2005

Richard A. Hinson

Gregory A. Nelson, Registration No. 30,577
Richard A. Hinson, Registration No. 47,652
Marc A. Boillot, Registration No. 56,164
AKERMAN SENTERFITT
Customer No. 40987
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 653-5000

Foundations of Statistical Natural Language Processing

Christopher D. Manning
Hinrich Schütze

The MIT Press
Cambridge, Massachusetts
London, England

Typeset in 10/13 Lucida Bright by the authors using LaTeX2e.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Information

Manning, Christopher D.
Foundations of statistical natural language processing / Christopher D. Manning, Hinrich Schütze.
p. cm.
Includes bibliographical references (p.) and index.
ISBN 0-262-13360-1
1. Computational linguistics—Statistical methods. I. Schütze, Hinrich. II. Title.
P98.5.S83M36 1999
410'.285—dc21 99-21137
CIP

15.1 Some Background on Information Retrieval

The goal of IR research is to develop models and algorithms for retrieving information from document repositories, in particular, textual information. The classical problem in IR is the *ad-hoc retrieval problem*. In ad-hoc retrieval, the user enters a query describing the desired information. The system then returns a list of documents. There are two main models. *Exact match* systems return documents that precisely satisfy some structured query expression, of which the best known type is *Boolean queries*, which are still widely used in commercial information systems. But for large and heterogeneous document collections, the result sets of exact match systems usually are either empty or huge and unwieldy, and so most recent work has concentrated on systems which rank documents according to their estimated relevance to the query. It is within such an approach that probabilistic methods are useful, and so we restrict our attention to such systems henceforth.

AD-HOC RETRIEVAL PROBLEM

EXACT MATCH

BOOLEAN QUERIES

An example of ad-hoc retrieval is shown in figure 15.1. The query is '"glass pyramid" Pei Louvre,' entered on the internet search engine Alta Vista. The user is looking for web pages about I. M. Pei's glass pyramid over the Louvre entrance in Paris. The search engine returns several relevant pages, but also some non-relevant ones – a result that is typical for ad-hoc searches due to the difficulty of the problem.

Some of the aspects of ad-hoc retrieval that are addressed in IR research are how users can improve the original formulation of a query interactively, by way of *relevance feedback*; how results from several text databases can be merged into one result list (*database merging*); which models are appropriate for partially corrupted data, for example, OCRed documents; and how the special problems that languages other than English pose can be addressed in IR.

RELEVANCE FEEDBACK

DATABASE MERGING

Some subfields of information retrieval rely on a training corpus of documents that have been classified as either relevant or non-relevant to a particular query. In *text categorization*, one attempts to assign documents to two or more pre-defined categories. An example is the subject codes assigned by Reuters to its news stories (Lewis 1992). Codes like CORP-NEWS (corporate news), CRUDE (crude oil) or ACQ (acquisitions) make it easier for subscribers to find stories of interest to them. A financial analyst interested in acquisitions can request a customized newsfeed that only delivers documents tagged with ACQ.

TEXT CATEGORIZATION

Filtering and *routing* are special cases of text categorization with only

FILTERING

ROUTING